PRÁCTICA 3
BAR DE TAPAS

Algorítmica

Grado en Ingeniería Informática
Universidad de granada

ADRA SANCHEZ RUIZ
CRISTINA GARRIDO AMADOR
JUAN MANUEL CASTILLO NIEVAS
LUIS LIÑÁN VILLAFRANCA
MARIANA ORIHUELA CAZORLA GRUPO A2

19 de abril de 2017, Granada

Índice

1. Presentación del problema

En esta práctica se nos ha asignado el problema del bar de tapas para desarrollarlo a través de un algoritmo voraz. Dicha práctica trata de resolver el siguiente enunciado:

Un restaurante de tapas ofrece un amplio surtido de n tapas, todas al mismo precio. De cada tapa t_i se conoce el número de calorías c_i que contiene. Se desea elegir como menú un subconjunto de tapas (sin repetir ninguna) que garantice un consumo de un mínimo de M calorías, pagando el menor precio posible. Diseñad un algoritmo voraz para resolver el problema y demostrad su optimalidad.

2. Pseudocódigo del algoritmo greedy

Candidatos: *tapas que nos ofrece el bar*
Función solución: *ya hemos consumido un número mínimo de calorías M*
Función selección: *seleccionar la tapa con mayor calorías*
Función de factibilidad: *siempre será factible nuestra solución, si no se ha alcanzado mínimo de calorías, siempre hay que insertar otra tapa*
Función de objetivo: *minimizar el número de tapas consumidas*

Algorithm 1 Algoritmo greedy

```
function MaximizadorDeCalorías(vector_tapas)
    Ordenar el vector_tapas de menor a mayor numero de calorías

    for i ← tamaño_vector_tapas − 1 to 0 do
        vector_solucion.add ← vector_tapas_i
        suma ← suma + vector_tapas_i.calorias

        if suma ≥ M then
            break

    return vector_solucion
```

3. Demostración de la optimalidad por reducción a lo absurdo

Tenemos un conjunto $S = t_0 > t_1 > \ldots > t_p$ que es una solución NO óptima. Sobre todas las soluciones óptimas $F = f_0 > f_1 > \ldots f_q$, es una de estas soluciones y $q < p$.

Llamamos r al máximo valor posible donde desde $f_0 = t_0, f_1 = t_1 \ldots f_r = t_r$.

Tenemos que $t_{(r+1)} < f_{(r+1)}$ porque así es como el algoritmo greedy lo selecciona. Entonces otra solución óptima del problema sería:

$L = t_0 > t_1 \ldots > t_r > t_{(r+1)} > f_{(r+2)} > \ldots > f_q$

Pero entonces alcanzaríamos una contradicción porque r no sería el máximo donde se alcanza la igualdad entre S y F, es decir, podrías coger un r mayor en la solución óptima.

4. Salida del algoritmo

Aquí mostramos algunas salidas por pantalla de nuestro algoritmo:

```
$ ./algoritmo_greedy 10 500
Las calorías de las tapas son: 68 291 50 286 202 187 284 465 472 151
Las calorías que vamos a tomar son: 472 465

$ ./algoritmo_greedy 10 1000
Las calorías de las tapas son: 5 404 149 300 423 272 235 194 418 368
Las calorías que vamos a tomar son: 423 418 404

$ ./algoritmo_greedy 10 1500
Las calorías de las tapas son: 400 222 11 415 3 327 256 153 220 250
Las calorías que vamos a tomar son: 415 400 327 256 250

$ ./algoritmo_greedy 10 2000
Las calorías de las tapas son: 392 422 485 358 295 284 113 97 328 153
Las calorías que vamos a tomar son: 485 422 392 358 328 295
```